

**ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД УКООПСПІЛКИ
«ПОЛТАВСЬКИЙ УНІВЕРСИТЕТ ЕКОНОМІКИ І ТОРГІВЛІ»**

**НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ БІЗНЕСУ
ТА СУЧАСНИХ ТЕХНОЛОГІЙ**

ФОРМА НАВЧАННЯ ЗАОЧНА

**КАФЕДРА МАТЕМАТИЧНОГО МОДЕЛЮВАННЯ ТА СОЦІАЛЬНОЇ
ІНФОРМАТИКИ**

Допускається до захисту

Завідувач кафедри _____ О.О. Ємець
(підпис)

«_____» _____ 2021 р.

**ПОЯСНЮВАЛЬНА ЗАПИСКА
ДО БАКАЛАВРСЬКОЇ РОБОТИ**

на тему

ТРЕНАЖЕР З ТЕМИ

**«Матриці суміжності для орієнтованих та неорієнтованих графів без петель»
дистанційного навчального курсу «Алгоритми та структури даних» та розробка
його програмного забезпечення.**

зі спеціальності 122 «Комп'ютерні науки та інформаційні технології»

Виконавець роботи Шабоян Артур Тігранович _____ «__» _____ 2021 р.
(підпис)

Науковий керівник к.ф.-м.н., проф., Ємець Єлизавета Михайлівна
_____ «__» _____ 2021 р.
(підпис)

ПОЛТАВА 2021 р.

РЕФЕРАТ

Записка: 45 с., 53 рис., 2 табл., 4 додатки (на 49 сторінках), 7 джерел.

Предмет розробки – тренажер з теми «Матриці суміжності для неорієнтованих графів без петель».

Мета роботи – розробка тренажеру, який буде перевіряти знання з теми «Матриці суміжності для неорієнтованих графів без петель» та навчати побудові матриць суміжності.

Методи розробки – мова програмування C++, середовище програмування Borland Builder 5.

Створено два алгоритми тренажеру та блок-схему.

Створено програму «Тренажер з теми «Матриці суміжності для орієнтованих та неорієнтованих графів без петель», яка складається з двох прикладів.

Ключові слова: НЕОРІЄНТОВАНИЙ ГРАФ, ОРІЄНТОВАНИЙ ГРАФ, МАТРИЦЯ СУМІЖНОСТІ, ТРЕНАЖЕР.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ	6
ВСТУП	7
1. ПОСТАНОВКА ЗАДАЧІ	8
2. ІНФОРМАЦІЙНИЙ ОГЛЯД	9
2.1. Огляд розробок, аналогічних темі випускової роботи	9
2.2. Позитивні аспекти розглянутих розробок	19
2.3. Недоліки розглянутих розробок	20
2.4. Необхідність та актуальність теми	20
3. ТЕОРЕТИЧНА ЧАСТИНА	21
3.1. Алгоритм тренажеру	21
3.2. Блок-схема алгоритму	31
4. ПРАКТИЧНА ЧАСТИНА	33
4.1. Опис програмної реалізації	34
4.2. Інструкція по роботі з програмою	34
ВИСНОВКИ	44
СПИСОК ЛІТЕРАТУРНИХ ДЖЕРЕЛ	45
ДОДАТОК А. ПРИКЛАД 2: АЛГОРИТМ	46
ДОДАТОК Б. БЛОК-СХЕМА	53
ДОДАТОК В. ПРОДОВЖЕННЯ ІНСТРУКЦІЇ	57
ДОДАТОК Г. ПРОГРАМНИЙ КОД	75

**ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ,
ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ**

Умовні позначення, символи, одиниці, скорочення, терміни	Пояснення умовних позначень, символів, одиниць, скорочень, термінів
n	Кількість вершин графа.
m	Кількість ребер графа.
v_i	Вершини графа з номером i .
(v_i, v_j)	Ребро графа, яке сполучає вершини v_i та v_j .
A	Матриця суміжності.
a_{ij}	Елемент матриці суміжності, що стоїть у рядку з номером i та у стовпці номером j .
$a_{ij} = 1$	Елемент матриці суміжності, значення якого вказує, що вершини v_i та v_j з'єднані ребром.
$a_{ij} = 0$	Елемент матриці суміжності, значення якого вказує, що вершини v_i та v_j не з'єднані ребром.
$a_{ii} = 1$	Елемент матриці суміжності, значення якого вказує, що наявна петля для вершини v_i .
$a_{ii} = 0$	Елемент матриці суміжності, значення якого вказує, що відсутня петля для вершини v_i .

ВСТУП

Тема бакалаврської роботи є *актуальною*, оскільки на даному етапі розвитку суспільства є потреба у кваліфікованих ІТ-кадрів, що у свою чергу призводить до необхідності постійного навчання та підвищення кваліфікації ІТ-розробників. Частина з них отримує освіту самостійно, за допомогою он-лайн ресурсів. А тренажери допоможуть таким студентам засвоїти нові знання.

Об'єктом розробки виступає тренажер.

Предметом розробки є тренажер з теми «Матриці суміжності для неорієнтованих та орієнтованих графів без петель».

Метою бакалаврської роботи є розробка програмного продукту – тренажеру, який буде перевіряти знання з теми «Матриці суміжності для неорієнтованих та орієнтованих графів без петель» та навчати побудові матриць суміжності.

Задачі бакалаврської роботи: знайомство з предметною областю; підготовка або розробка прикладів для тренажеру; створення алгоритму тренажера; розробка програми; тестування програми та створення документації.

Методами розробки є мова програмування C++, середовище програмування Borland Builder 5.

Готовність до впровадження – програмний продукт повністю є готовим.

Пояснювальна записка складається з чотирьох частин: постановки задачі, інформаційного огляду, теоретичної частини, практичної частини.

У постановці задачі описано задачі дипломного проектування та вимоги до програми. У інформаційному огляді проаналізовано три розробки з заданою теми. У теоретичній частині викладено приклади та сформульовано алгоритми тренажеру, висвітлено блок-схему алгоритму. У практичній частині пояснено створений програмний код та роботу програмного продукту.

Пояснювальна записка має *обсяг* 45 сторінок. В цей обсяг не включені додатки.

1. ПОСТАНОВКА ЗАДАЧІ

Необхідно ознайомитись теоретичними відомостями тем «Графи: основні означення», «Способи представленнями графів».

Особливу увагу зосередити на представленні графів за допомогою матриць суміжності.

Взяти з літератури, дистанційного курсу або розробити самостійно два приклади, в яких на базі неорієнтованого графу без петель (приклад 1) та орієнтованого графу без петель (приклад 2) будуються матриці суміжності.

Позначки і означення повинні бути такими ж як у відповідній темі дистанційного курсу «Алгоритми і структури даних» Полтавського університету економіки і торгівлі.

Для підготовлених прикладів розробити алгоритми тренажеру. В тренажері також слід перевіряти теоретичні знання студентів.

Побудувати блок-схеми алгоритмів.

Створити програму на базі розроблених алгоритмів.

Перевірити програмний продукт на працездатність.

Створити документацію по програмі.

2. ІНФОРМАЦІЙНИЙ ОГЛЯД

2.1. Огляд розробок, аналогічних темі випускової роботи

1) *Сайт університету Сан-Франциско* (адреса <http://www.cs.usfca.edu/~galles/visualization/BFS.html>) дає можливість ознайомитись з представленням графів у графічному вигляді та у вигляді матриці суміжності (рис. 2.1).

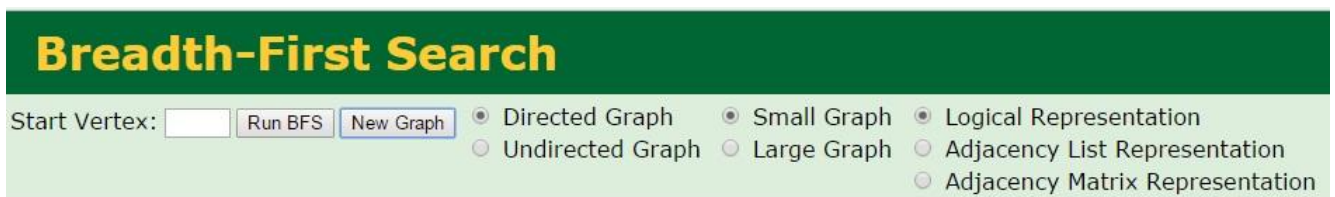


Рисунок 2.1 – Візуалізація

Пояснимо роботу візуалізації (рис. 2.1).

Кнопка «New Graph» призначена для генерування нового графу.

Перемикач «Small Graph» призначений для задання графу з 8 вершинами (рис. 2.2), «Large Graph» – з 18 вершинами.

Перемикач «Directed Graph» – задає орієнтований граф (рис. 2.4), «Undirected Graph» – неорієнтований (рис. 2.2).

Перемикач «Logical Representation» – задає граф у вигляді рисунку (рис. 2.2, 2.4), «Adjacency List Representation» – демонструє той же граф списком суміжності, «Adjacency Matrix Representation» – показує той же граф матрицею суміжності (рис. 2.3, 2.5).

2) *Тренажер за темою «Матриці суміжності та інцидентності»*, створений магістранткою спеціальності «Соціальна інформатика» ПУЕТ Іваховою Юлією у 2017 р. для дистанційного курсу ПУЕТ «Дискретна математика» (рис. 2.6).

Сімулятор містить 10 різнопланових прикладів (рис. 2.7). Побудові матриці суміжності за графом та навпаки присвячено 4 задачі (рис. 2.7-2.14).

- ☒ Small Graph ☒ Logical Representation
- ☐ Large Graph ☐ Adjacency List Representation
- ☐ Adjacency Matrix Representation

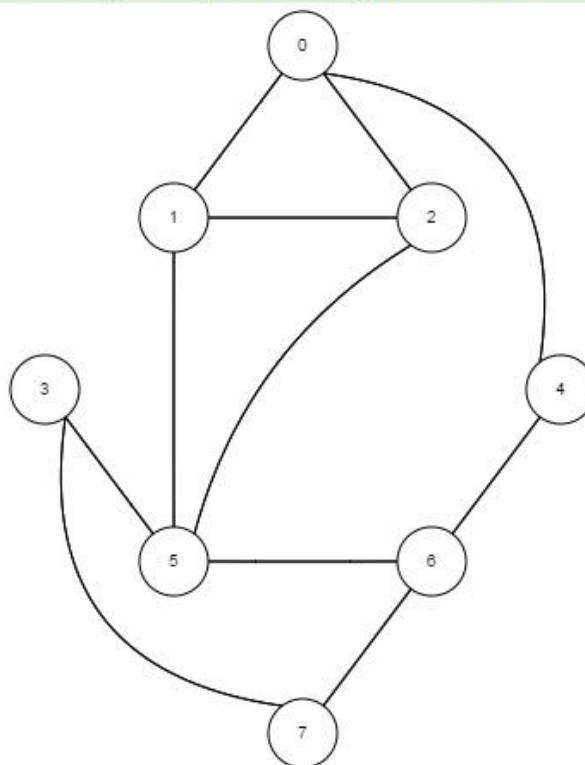


Рисунок 2.2 – Згенерований неорієнтований граф

- ☒ Small Graph ☐ Logical Representation
- ☐ Large Graph ☐ Adjacency List Representation
- ☒ Adjacency Matrix Representation

	0	1	2	3	4	5	6	7
0		1	1		1			
1	1		1			1		
2	1	1				1		
3						1		1
4	1						1	
5		1	1	1			1	
6					1	1		1
7				1			1	

Рисунок 2.3 – Граф з рис. 2.2, представлений матрицею суміжності

- Small Graph
- Logical Representation
- Large Graph
- Adjacency List Representation
- Adjacency Matrix Representation

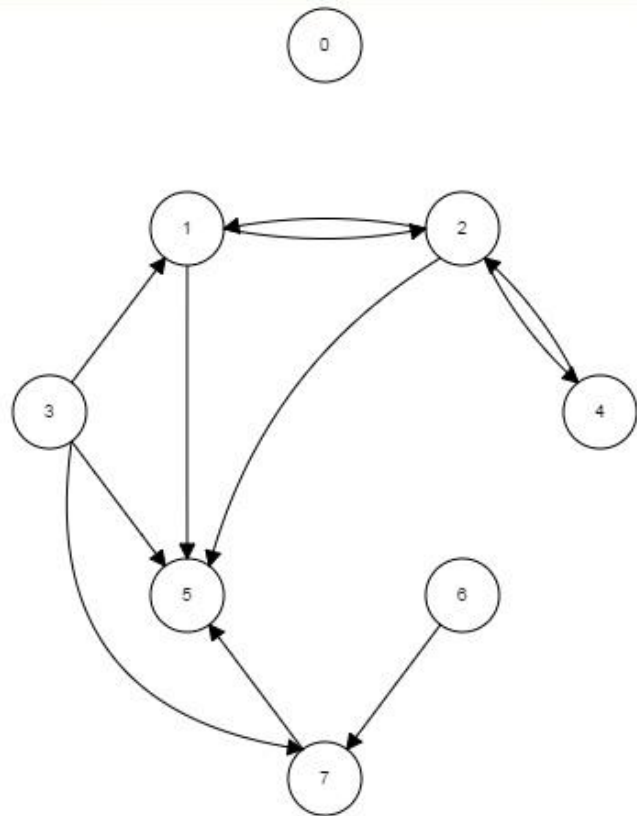


Рисунок 2.4 – Згенерований орієнтований граф

- Small Graph
- Logical Representation
- Large Graph
- Adjacency List Representation
- Adjacency Matrix Representation

	0	1	2	3	4	5	6	7
0								
1			1			1		
2		1			1	1		
3		1				1		1
4			1					
5								
6								1
7						1		

Рисунок 2.5 – Граф з рис. 2.4, представлений матрицею суміжності



Рисунок 2.6 – Тренажер за темою «Матриці суміжності та інцидентності»

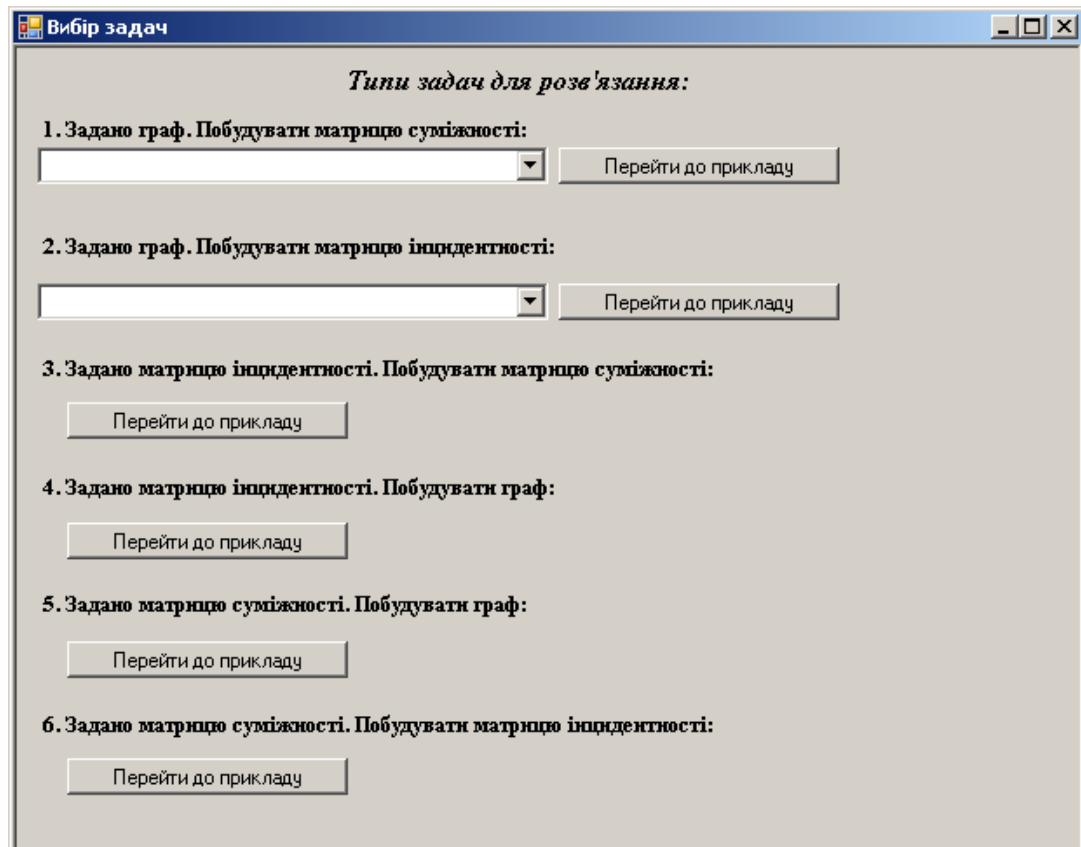


Рисунок 2.7 – Меню тренажеру

1. Задано граф. Побудувати матрицю суміжності:

заданий неорієнтований граф

заданий неорієнтований граф з ізольованою вершиною

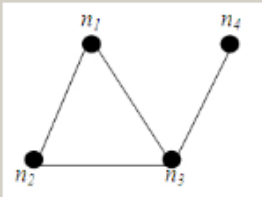
заданий орієнтований граф з петлями

Перейти до прикладу

Рисунок 2.8 – Вибір прикладів

Побудова матриці суміжності за заданим неорієнтованим графом

Для заданого неорієнтованого графа побудувати [матрицю суміжності](#)



Чи є матриця суміжності квадратною?

☒ Так
 ☐ Ні

Яка вимірність матриці?

4

Чи симетричною є матриця суміжності відносно головної діагоналі?

☒ Так
 ☐ Ні

Побудова матрицю суміжності

0	1	1	0
1	0	1	0
1	1	0	1
0	0	1	0

Рівень засвоєння матеріалу складає 100 %

Повернутися в меню

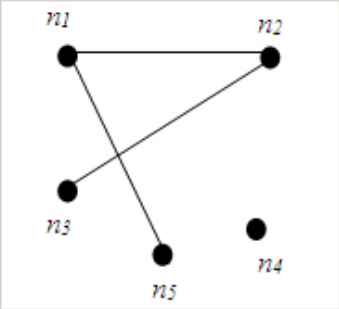
Наступний приклад

Рисунок 2.9 – Тренінг з побудови матриці суміжності за неорієнтованим графом

Тренажер містить необхідні теоретичні відомості та розрахункові приклади, які можна побачити клацнувши на виділений синім кольором термін (див.. наприклад, рис. 2.11). Натиснувши кнопку «Закрити», ці викладки можна згорнути (рис. 2.11).

Побудова матриці суміжності за заданим графом з ізольованою вершиною

Для заданого неорієнтованого графа побудувати [матрицю суміжності](#)



Чи є матриця суміжності квадратною?

☒ Так

☐ Ні

Яка вимірність матриці?

5

Чи симетричною є матриця суміжності відносно головної діагоналі?

☒ Так

☐ Ні

Побудова матриці суміжності

0	1	0	0	1
1	0	1	0	0
0	1	0	0	0
0	0	0	0	0
1	0	0	0	0

Рівень засвоєння матеріалу складає 94 %

Повернутися в меню

Наступний приклад

Рисунок 2.10 – Тренінг з побудови матриці суміжності
за неорієнтованим графом із ізольованою вершиною

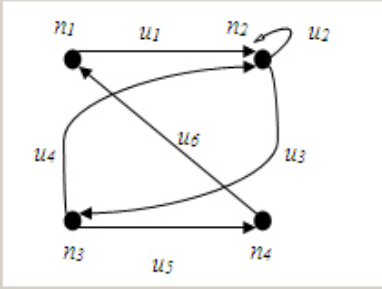
Червоним кольором (див., наприклад, рис. 2.10) у прикладах виділяються надані помилкові відповіді, які після пояснення помилки (рис. 2.12) програмою автоматично виправляються. Але поле, де була помилка залишається червоного кольору.

Зеленим кольором (див., наприклад, рис. 2.10) виділяються вірні відповіді.

Жовтим кольором позначено, куди вводити чергову відповідь (рис. 2.11).

Побудова матриці суміжності за заданим орієнтованим графом

Для заданого орієнтованого графа побудувати матрицю суміжності



Чи є матриця суміжності квадратною?

☒ Так

☐ Ні

Яка вимірність матриці?

4

Чи містить даний граф петлі?

☒ Так

☐ Ні

Чи симетричною є матриця суміжності відносно головної діагоналі?

☐ Так

☒ Ні

Побудова матриці суміжності

0	1	1	0

Визначення

Матрицею суміжності називають квадратну матрицю $R = (r_{ij})$ порядку $k \times k$, у якій вершини графа n_1, \dots, n_k та зв'язки між вершинами позначаються наступним чином:

$$r_{ij} = \begin{cases} 1, & \text{якщо є дуга } (n_i, n_j), \\ 0, & \text{якщо такої дуги немає.} \end{cases}$$

Закрити

Рисунок 2.11– Тренінг з побудови матриці суміжності за орієнтованим графом

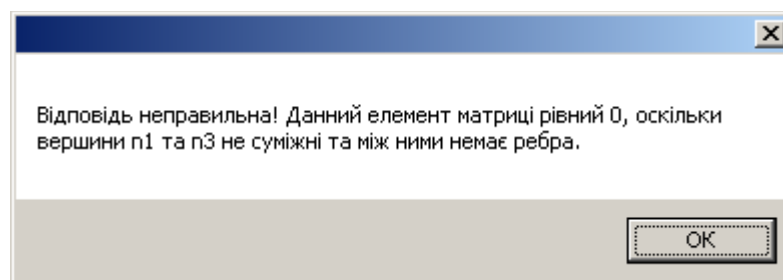


Рисунок 2.12 – Обробка помилки (продовження тренінгу з рис. 2.11)

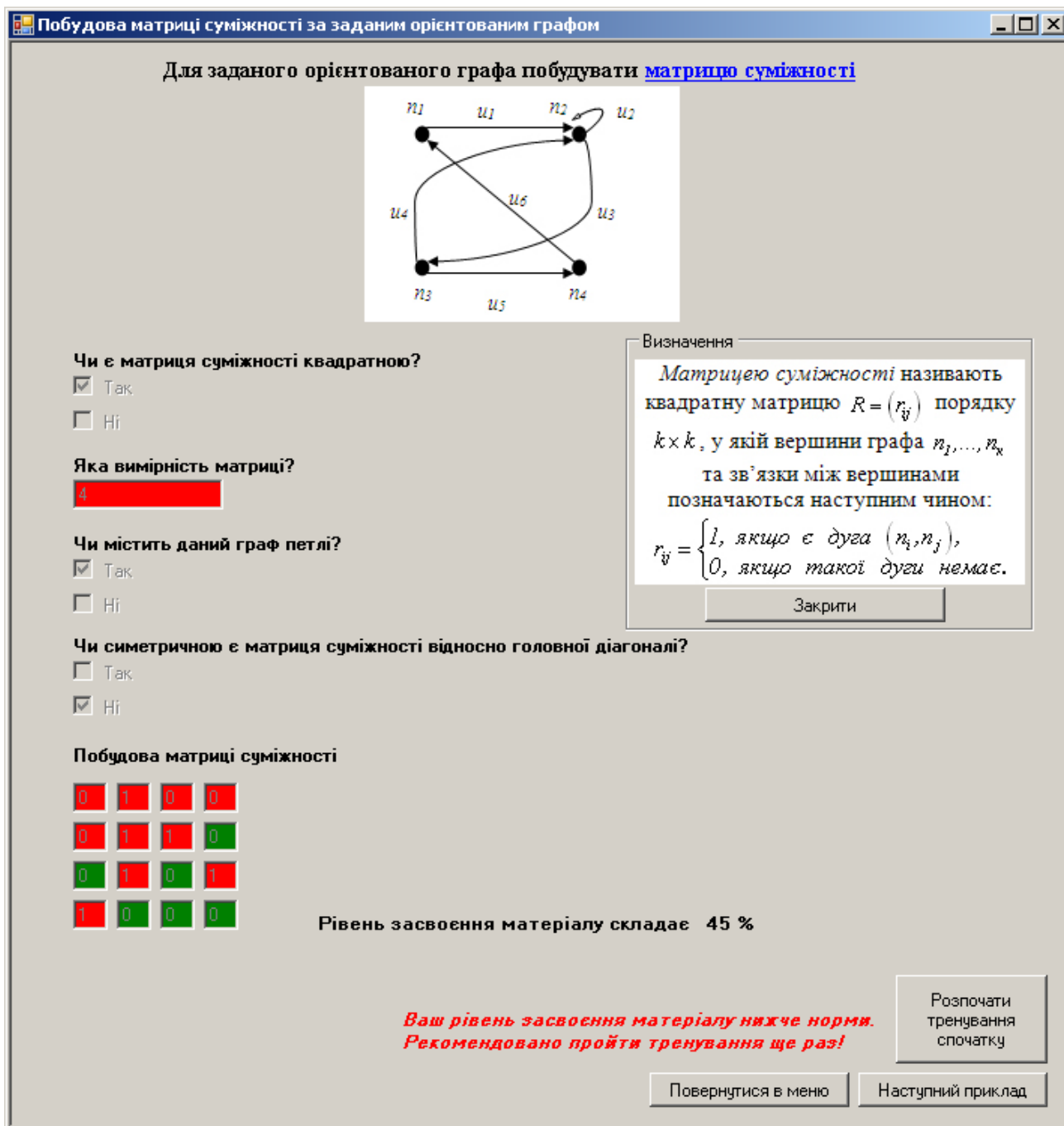


Рисунок 2.13 – Тренінг з побудови матриці суміжності за орієнтованим графом (продовження рис. 2.11)

Після проходження тренінгу подається у відсотках рівень засвоєння матеріалу. У випадку низьких відсотків надається рекомендація з повтореного проходження тренінгу (рис. 2.13).

Є можливість повторного проходження того ж самого прикладу (див., наприклад, рис. 2.11), повернення до меню (рис. 2.11) або переходу до наступного завдання (рис. 2.11).

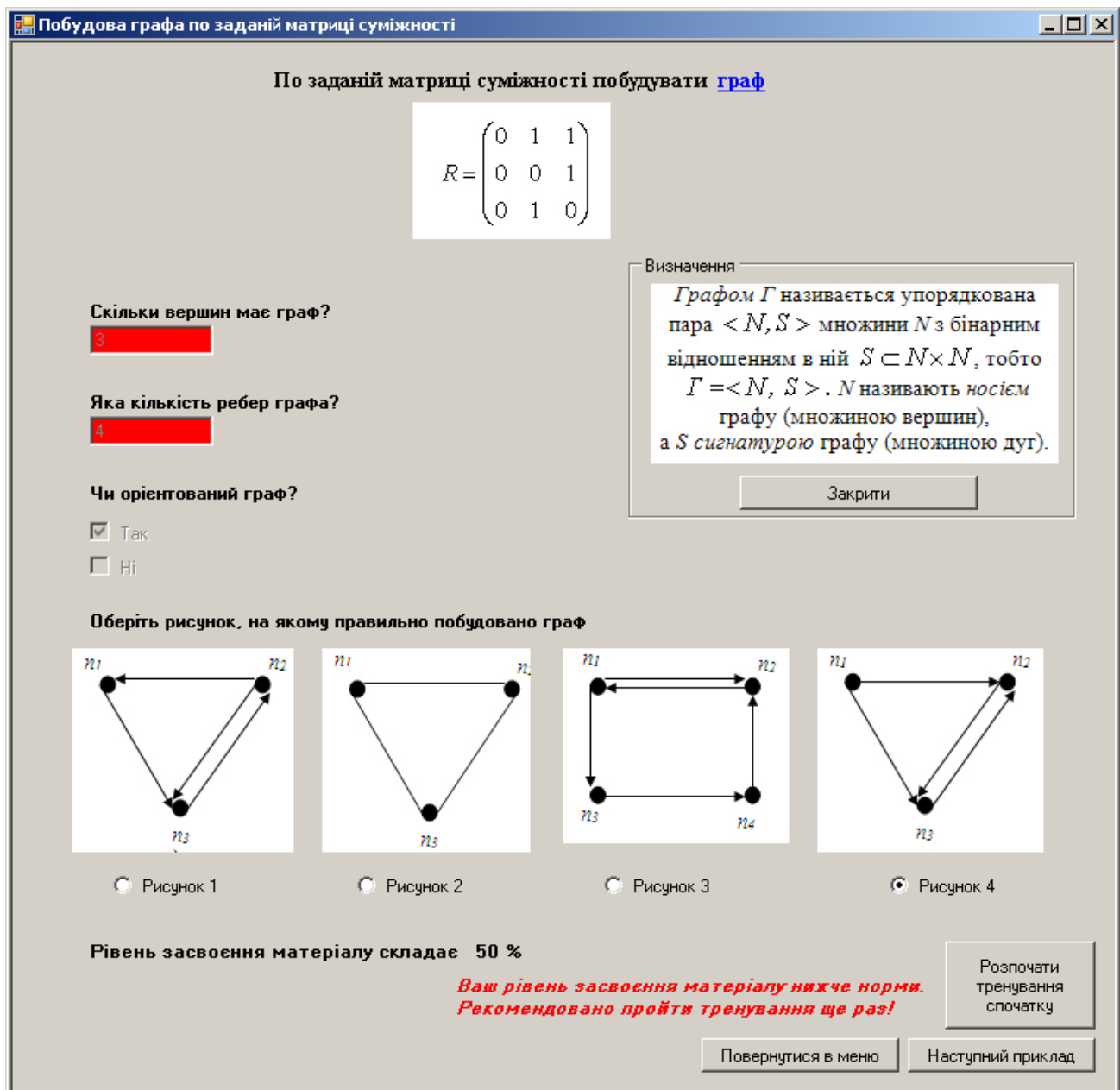


Рисунок 2.14 – Тренінг з побудови графа за матрицею суміжності

3) *Ресурс «Graph Online» https://graphonline.ru/create_graph_by_matrix* (рис. 2.15) дає можливість побудувати граф за заданою матрицею суміжності (рис. 2.16-2.19) та навпаки (рис. 2.20).

Якщо матриця суміжності задається несиметричною (рис. 2.16), то будується орієнтований граф (рис. 2.17), інакше (рис. 2.18) – неорієнтований (рис. 2.19).

Є відеосправка (рис. 2.15), можливість зміни мови з російської на англійську (рис. 2.15) та багато інших функцій (рис. 2.15).

Построение графа по матрице смежности

На этой странице вы можете задать матрицу смежности и построить по ней граф

Задайте матрицу смежности. Затем нажмите кнопку "Построить граф".

Ввести в таблицу

Ввести как текст

✓ Построить граф

0	1	0
1	0	0
0	1	0

+ Добавить вершину в матрицу

Видео справка Wiki справка

Построение графа по матрице смежности

На этой странице вы можете задать матрицу смежности и построить по ней граф

Задайте матрицу смежности. Затем нажмите кнопку "Построить граф".

Ввести в таблицу

Ввести как текст

✓ Построить граф

0	1	0
1	0	0
0	1	0

+ Добавить вершину в матрицу

Видео справка Wiki справка

Рисунок 2.15 – Ресурс «Graph Online»

Ввести в таблицу

Ввести как текст

0	1	0
1	0	0
0	1	0

+ Добавить вершину в матрицу

Рисунок 2.16 – Матрица суміжності (симетрична)

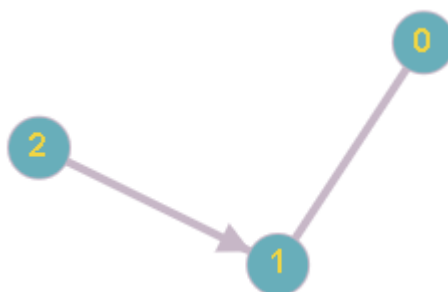


Рисунок 2.17 – Граф для матриці суміжності з рис. 2.16

Ввести в таблицу

Ввести как текст

0	1	0
1	0	1
0	1	0

+ Добавить вершину в матрицу

Рисунок 2.18 – Матриця суміжності (симетрична)

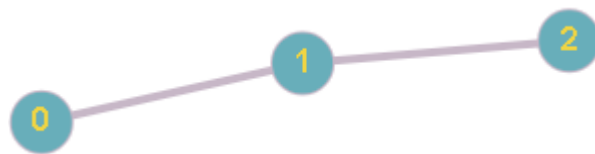


Рисунок 2.19 – Граф для матриці суміжності з рис. 2.18

Работа с графами онлайн

Визуализация графа, поиск кратчайшего пути и многое другое. В разделе Справка вы найдете обучающие видео.

Граф

Вид

По умолчанию

+ Добавить вершину

Соединить вершины

Алгоритмы

Удалить

Настройки

Кликните на рабочую область, чтобы добавить вершину.

Нумерация вершин 1, 2, 3...

5

1

7

0

6

4

3

2

Матрица смежности

Задайте матрицу смежности. Используйте запятую "," в качестве разделителя

0, 1, 1, 1, 0, 0, 0, 0,
 1, 0, 1, 0, 0, 1, 0, 1,
 1, 1, 0, 1, 1, 0, 0, 1,
 1, 0, 1, 0, 1, 1, 0, 0, 0,
 0, 0, 1, 1, 0, 0, 1, 1,
 0, 1, 0, 0, 0, 0, 1, 1,
 0, 0, 0, 0, 1, 1, 0, 1,
 0, 1, 1, 0, 1, 1, 1, 0,

Сохранить

Отмена

Рисунок 2.20 – Побудова матриці за графом

2.2. Позитивні аспекти розглянутих розробок

Розробка №1 – можливість генерації різних прикладів. великого та малого розміру; зручний інтерфейс; приємний дизайн.

Розробка №2 – велика кількість різноманітних прикладів; наявність теорії; виділення кольором; пояснення помилок; підрахунок рівня засвоєння матеріалу; зручна навігація; приємний дизайн та зрозумілий інтерфейс.

Розробка №3 – наявність відеодовідки; двомовний інтерфейс; задання матриці двома способами як тексту та як таблиці; можливість зміни масштабу рисунку графа; можливість зміни розташування вершин графу, переміщення графу, збереження та друку графу та багато іншого функціоналу.

2.3. Недоліки розглянутих розробок

Розробки №1 та **№3** лише ілюстративні анімації. Вони не проводять тренінгу студента, а лише дають можливість краще, наочно продемонструвати матеріал.

Розробка №2 недоліків немає і повністю задовольняє вимогам дистанційного навчання.

2.4. Необхідність та актуальність теми

Не дивлячись на наявність візуалізацій, слід відмітити, що вони, по-перше, представлені іноземними мовами, по-друге, являються лише демонстраціями, не тренують студента із заданої теми.

Тому, створення тренажера з українськомовним інтерфейсом з прийнятими в дистанційному курсі позначками і термінологією є актуальною та необхідною задачею при дистанційному навчанні.

3. ТЕОРЕТИЧНА ЧАСТИНА

3.1. Алгоритм тренажеру

В програмі умова прикладу завжди повинна бути видима.

Необхідна інформація з попередніх кроків також повинна бути доступна або видима.

При неправильній відповіді з'являється повідомлення, в якому пояснюється помилка. Користувачу дається можливість знову надати відповідь.

При правильній відповіді відбувається перехід на наступний крок алгоритму.

Приклад 1.

Побудова матриця суміжності для неорієнтованого графа.

Умова. Задати граф (рис. 3.1) матрицею суміжності.

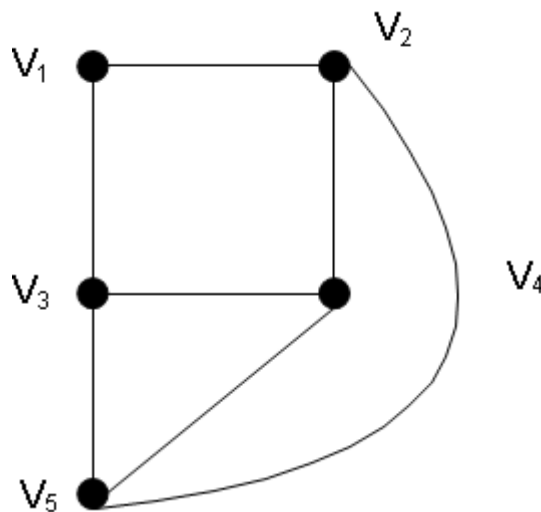


Рисунок 3.1 – Граф

1. Граф, зображений на рисунку, є

- орієнтованим.
- неорієнтованим.
- змішаним.

Вірна відповідь – «неорієнтованим.».

При помилці з'являється пояснення «На ребрах графа не вказано напрямки руху, отже, цей граф є неорієнтованим.».

2. Скільки вершин у графі (рис. 3.2)?

$n =$

Рисунок 3.2 – Форма вводу

Вірна відповідь – «5».

При помилці з'являється пояснення «В графі 5 вершин.».

3. Скільки ребер у графі (рис. 3.3)?

$m =$

Рисунок 3.3 – Форма вводу

Вірна відповідь – «7».

При помилці з'являється пояснення «В графі 7 ребер.».

4. Кількість ребер графа впливає на вимірність матриці суміжності?

- Так.
- Ні.

Вірна відповідь – «Ні.».

При помилці з'являється пояснення «Кількість ребер графа не впливає на вимірність матриці суміжності.».

5. Матриця суміжності квадратна?

- Так.
- Ні.

Вірна відповідь – «Так.».

При помилці з'являється пояснення «В матриці суміжності кількість стовпців дорівнює кількості рядків. Отже, матриця квадратна.».

6. Яка вимірність матриці суміжності (рис. 3.4)?



Рисунок 3.4 – Форма вводу

Вірна відповідь – «5 x 5».

При помилці з'являється пояснення «В матриці суміжності кількість рядків і кількість стовпців дорівнюють кількості вершин графа. Отже, вимірність матриці 5 x 5.».

7. Матриця суміжності однозначно задає граф?

- Так.
- Ні.

Вірна відповідь – «Так.».

При помилці з'являється пояснення «Будь-який граф однозначно задається матрицею суміжності.».

8. Елементи матриці суміжності a_{ij} можуть приймати значення (можна обрати декілька вірних відповідей):

- -1.
- 0.
- 1.
- будь-які числа.

Вірні відповіді – «0; 1.».

При помилці з'являється пояснення «Елементи матриці суміжності дорівнюють нулю або одиниці.».

9. Для елемента матриці суміжності a_{ij} справедливе твердження (можна обрати декілька вірних відповідей) :

- $a_{ij} = 1$, якщо вершини v_i та v_j сполучені ребром.
- $a_{ij} = 0$, якщо вершини v_i та v_j сполучені ребром.
- $a_{ij} = 1$, якщо вершини v_i та v_j не поєднуються ребром.
- $a_{ij} = 0$, якщо вершини v_i та v_j не поєднуються ребром.

Вірні відповіді – « $a_{ij} = 1$, якщо вершини v_i та v_j сполучені ребром; $a_{ij} = 0$, якщо вершини v_i та v_j не поєднуються ребром.».

При помилці з'являється пояснення « $a_{ij} = 1$, якщо вершини v_i та v_j сполучені ребром. $a_{ij} = 0$, якщо вершини v_i та v_j не поєднуються ребром.».

10. Заповніть матрицю суміжності (рис. 3.5):

$A =$

	v_1	v_2	v_3	v_4	v_5
v_1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
v_2	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
v_3	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
v_4	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
v_5	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Рисунок 3.5 – Форма вводу

Матриця заповнюється по рядкам зліва направо.

Перший рядок (рядок v_1).

Перша клітинка (на перетині рядка v_1 і стовпця v_1).

Вірна відповідь – «0».

При помилці з'являється пояснення «Вершина v_1 немає петлі, тому елемент матриці $a_{11} = 0$.».

Друга клітинка (на перетині рядка v_1 і стовпця v_2).

Вірна відповідь – «1».

При помилці з'являється пояснення «Вершини v_1 і v_2 з'єднані ребром, тому елемент матриці $a_{12} = 1$ ».

Третя клітинка (на перетині рядка v_1 і стовпця v_3).

Вірна відповідь – «1».

При помилці з'являється пояснення «Вершини v_1 і v_3 з'єднані ребром, тому елемент матриці $a_{13} = 1$ ».

Четверта клітинка (на перетині рядка v_1 і стовпця v_4).

Вірна відповідь – «0».

При помилці з'являється пояснення «Вершини v_1 і v_4 не з'єднані ребром, тому елемент матриці $a_{14} = 0$ ».

П'ята клітинка (на перетині рядка v_1 і стовпця v_5).

Вірна відповідь – «0».

При помилці з'являється пояснення «Вершини v_1 і v_5 не з'єднані ребром, тому елемент матриці $a_{15} = 0$ ».

Другий рядок (рядок v_2).

Перша клітинка (на перетині рядка v_2 і стовпця v_1).

Вірна відповідь – «1».

При помилці з'являється пояснення «Вершини v_2 і v_1 з'єднані ребром, тому елемент матриці $a_{21} = 1$ ».

Друга клітинка (на перетині рядка v_2 і стовпця v_2).

Вірна відповідь – «0».

При помилці з'являється пояснення «Вершина v_2 немає петлі, тому елемент матриці $a_{22} = 0$ ».

Третя клітинка (на перетині рядка v_2 і стовпця v_3).

Вірна відповідь – «0».

При помилці з'являється пояснення «Вершини v_2 і v_3 не з'єднані ребром, тому елемент матриці $a_{23} = 0$ ».

Четверта клітинка (на перетині рядка v_2 і стовпця v_4).

Вірна відповідь – «1».

При помилці з'являється пояснення «Вершини v_2 і v_4 з'єднані ребром, тому елемент матриці $a_{24} = 1$ ».

П'ята клітинка (на перетині рядка v_2 і стовпця v_5).

Вірна відповідь – «1».

При помилці з'являється пояснення «Вершини v_2 і v_5 з'єднані ребром, тому елемент матриці $a_{25} = 1$ ».

Третій рядок (рядок v_3).

Перша клітинка (на перетині рядка v_3 і стовпця v_1).

Вірна відповідь – «1».

При помилці з'являється пояснення «Вершини v_3 і v_1 з'єднані ребром, тому елемент матриці $a_{31} = 1$ ».

Друга клітинка (на перетині рядка v_3 і стовпця v_2).

Вірна відповідь – «0».

При помилці з'являється пояснення «Вершини v_3 і v_2 не з'єднані ребром, тому елемент матриці $a_{32} = 0$ ».

Третя клітинка (на перетині рядка v_3 і стовпця v_3).

Вірна відповідь – «0».

При помилці з'являється пояснення «Вершина v_3 немає петлі, тому елемент матриці $a_{33} = 0$ ».

Четверта клітинка (на перетині рядка v_3 і стовпця v_4).

Вірна відповідь – «1».

При помилці з'являється пояснення «Вершини v_3 і v_4 з'єднані ребром, тому елемент матриці $a_{34} = 1$.».

П'ята клітинка (на перетині рядка v_3 і стовпця v_5).

Вірна відповідь – «1».

При помилці з'являється пояснення «Вершини v_3 і v_5 з'єднані ребром, тому елемент матриці $a_{35} = 1$.».

Четвертий рядок (рядок v_4).

Перша клітинка (на перетині рядка v_4 і стовпця v_1).

Вірна відповідь – «0».

При помилці з'являється пояснення «Вершини v_4 і v_1 не з'єднані ребром, тому елемент матриці $a_{41} = 0$.».

Друга клітинка (на перетині рядка v_4 і стовпця v_2).

Вірна відповідь – «1».

При помилці з'являється пояснення «Вершини v_4 і v_2 з'єднані ребром, тому елемент матриці $a_{42} = 1$.».

Третя клітинка (на перетині рядка v_4 і стовпця v_3).

Вірна відповідь – «1».

При помилці з'являється пояснення «Вершини v_4 і v_3 з'єднані ребром, тому елемент матриці $a_{43} = 1$.».

Четверта клітинка (на перетині рядка v_4 і стовпця v_4).

Вірна відповідь – «0».

При помилці з'являється пояснення «Вершина v_4 немає петлі, тому елемент матриці $a_{44} = 0$.».

П'ята клітинка (на перетині рядка v_4 і стовпця v_5).

Вірна відповідь – «1».

При помилці з'являється пояснення «Вершини v_4 і v_5 з'єднані ребром, тому елемент матриці $a_{45} = 1$ ».

П'ятий рядок (рядок v_5).

Перша клітинка (на перетині рядка v_5 і стовпця v_1).

Вірна відповідь – «0».

При помилці з'являється пояснення «Вершини v_5 і v_1 не з'єднані ребром, тому елемент матриці $a_{51} = 0$ ».

Друга клітинка (на перетині рядка v_5 і стовпця v_2).

Вірна відповідь – «1».

При помилці з'являється пояснення «Вершини v_5 і v_2 з'єднані ребром, тому елемент матриці $a_{52} = 1$ ».

Третя клітинка (на перетині рядка v_5 і стовпця v_3).

Вірна відповідь – «1».

При помилці з'являється пояснення «Вершини v_5 і v_3 з'єднані ребром, тому елемент матриці $a_{53} = 1$ ».

Четверта клітинка (на перетині рядка v_5 і стовпця v_4).

Вірна відповідь – «1».

При помилці з'являється пояснення «Вершини v_5 і v_4 з'єднані ребром, тому елемент матриці $a_{54} = 1$ ».

П'ята клітинка (на перетині рядка v_5 і стовпця v_5).

Вірна відповідь – «0».

При помилці з'являється пояснення «Вершина v_5 немає петлі, тому елемент матриці $a_{55} = 0$ ».

11. Отримана матриця суміжності

$$A = \begin{matrix} & \begin{matrix} v_1 & v_2 & v_3 & v_4 & v_5 \end{matrix} \\ \begin{matrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{matrix} & \begin{pmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{pmatrix} \end{matrix}$$

симетрична відносно головної діагоналі?

- Так.
- Ні.

Вірна відповідь – «Так.».

При помилці з'являється пояснення «Оскільки, перший рядок матриці збігається з першим стовпцем матриці, другий рядок збігається з другим стовпцем, третій рядок – з третім стовпцем, четвертий рядок – з четвертим стовпцем, п'ятий рядок – з п'ятим стовпцем, то матриця є симетричною відносно головної діагоналі.».

12. Даний граф має петлі?

- Так.
- Ні.

Вірна відповідь – «Ні.».

При помилці з'являється пояснення «Петлею називається ребро, яке з'єднує одну й ту саму вершину. Таких ребер на рисунку немає. Отже, заданий граф без петель.».

13. Матриця суміжності для неорієнтованого графу без петель завжди симетрична відносно головної діагоналі?

- Так.
- Ні.

Вірна відповідь – «Так.».

При помилці з'являється пояснення «Матриця суміжності для неорієнтованого графу без петель завжди симетрична відносно головної діагоналі.».

14. На головній діагоналі отриманої матриці суміжності

$$A = \begin{matrix} & \begin{matrix} v_1 & v_2 & v_3 & v_4 & v_5 \end{matrix} \\ \begin{matrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{matrix} & \begin{pmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{pmatrix} \end{matrix}$$

лише нульові елементи?

- Так.
- Ні.

Вірна відповідь – «Так.».

При помилці з'являється пояснення « $a_{11} = a_{22} = a_{33} = a_{44} = a_{55} = 0$. Отже, на головній діагоналі даної матриці суміжності лише нульові елементи.».

15. Матриця суміжності для неорієнтованого графу без петель завжди містить лише нульові елементи на головній діагоналі?

- Так.
- Ні.

Вірна відповідь – «Так.».

При помилці з'являється пояснення «Оскільки, головна діагональ матриці суміжності показує наявність чи відсутність петель у графі, то для

неорієнтованого графу без петель на головній діагоналі будуть лише нульові елементи.».

З'являється повідомлення «Роботу з першим прикладом завершено!».

Алгоритм тренажера прикладу 2 подано в додатку А.

3.2. Блок-схема алгоритму

На рис. Б.1-Б.4 зображена блок-схема алгоритму першого прикладу (кроки 1-9). Інші кроки аналогічні.

4. ПРАКТИЧНА ЧАСТИНА

4.1. Опис програмної реалізації

Для створення програми було обрано середовище Borland Builder 5 та мову програмування C++.

Найбільш цікавим та складаним кроком тренажеру при програмній реалізації прикладу 1 став крок №10. Опишемо його.

На цьому етапі програми слід вводити елементи матриці суміжності. Тобто слід послідовно ввести 15 значень.

Користувач вводить значення у поле, виділене жовтим кольором, спочатку – у перше поле.

Всі поля, крім першого, заблоковані.

Для цього першого компонента для події втрата фокусу (OnExit), яка виникне тоді, коли користувач натисне кнопку «Перевірити» або буде рухатися по клітинкам за допомогою клавіші Tab, був написаний код (табл. 4.1):

Таблиця 4.1 – Код програми

Номер рядка	Код
1	<code>void __fastcall TForm1::Edit1Exit(TObject *Sender)</code>
2	<code>{</code>
3	<code> if (Edit1->Text=="")</code>
4	<code> {</code>
5	<code> MessageDlg("Надайте відповідь!", mtWarning,</code>
6	<code> TMsgDlgButtons() << mbOK, 0);</code>
7	<code> Edit1->SetFocus();</code>
8	<code> return;</code>
9	<code> }</code>
10	<code> if (Edit1->Text=="0")</code>
11	<code> {</code>
12	<code> MessageDlg("Відповідь вірна!", mtInformation,</code>
13	<code> TMsgDlgButtons() << mbOK, 0);</code>
14	<code> Edit2->Enabled=true;</code>
15	<code> Edit2->SetFocus();</code>
16	<code> Edit2->Color=clYellow;</code>
17	<code> Edit1->Color=clWindow;</code>
18	<code> Edit1->Enabled=false;</code>
19	<code> }</code>

13	<pre> } else { MessageDlg("Відповідь помилкова!\nВершина V1 немає петлі, тому елемент матриці A11 = 0.", mtError, TMsgDlgButtons() << mbOK, 0); Edit1->SetFocus(); } } </pre>
----	--

Тобто, якщо користувач залишить поле пустим (рядок 2), то на екрані буде повідомлення про це (рядок 3), курсор повернеться в це ж поле (рядок 4) і відбувається вихід з процедури (рядок 5).

Якщо користувач надасть вірну відповідь (тут це 0, рядок 6), то на екрані буде повідомлення про вірність відповіді (рядок 7), наступне текстове поле стане розблокованим (рядок 8), курсор буде стояти в наступному текстовому полі (рядок 9), колір фону цього поля стане жовтим (рядок 10).

Далі поточне текстове поле стане заблокованим (рядок 11) і його колір з жовтого зміниться на білий (рядок 12).

Якщо користувач надасть хибну відповідь (рядок 13), то на екрані буде пояснення помилки (рядок 14), курсор повернеться в це ж поле (рядок 15).

Для всіх інших текстових полів код аналогічний. Буде лише іншим пояснення помилки. Наприклад, для другого поля воно наступне:

```

MessageDlg("Відповідь помилкова!\nВершини V1 та V2 з'єднані
ребром, тому елемент матриці A12 = 1.", mtError, TMsgDlgButtons() <<
mbOK, 0);

```

Код трохи відрізняється для останнього текстового поля (табл. 4.2).

Різниця полягає в тому, що немає переходу до наступного текстового поля та налаштувань цього поля, а здійснюється перехід до наступного кроку (рядок 8) та приховування поточної форми (рядок 9).

Таблиця 4.2 – Код програми

Номер рядка	Код
1	void __fastcall TForm11::Edit25Exit(TObject *Sender)
2	{
3	if (Edit25->Text=="")
4	{
5	MessageDlg("Надайте відповідь!", mtWarning,
6	TMsgDlgButtons() << mbOK, 0);
7	Edit25->SetFocus();
8	return;
9	}
10	if (Edit25->Text=="0")
11	{
12	MessageDlg("Відповідь вірна!", mtInformation,
13	TMsgDlgButtons() << mbOK, 0);
14	Form12->Show();
15	Form11->Hide();
16	Form11->Edit25->Color=clWindow;
17	Form11->Edit25->Enabled=false;
18	}
19	else
20	{
21	MessageDlg("Відповідь помилкова!\nВершина V5 немає
22	петлі, тому елемент матриці A55 = 0.", mtError,
23	TMsgDlgButtons() << mbOK, 0);
24	Edit25->SetFocus();
25	}
26	}

Програмний код, створений для інших кроків, винесено у додаток Г. В зв'язку зі значним обсягом тексту, у додатку представлено лише типові кроки прикладу 1.

4.2. Інструкція по роботі з програмою

Робота програми показана на рисунках 4.1-4.28, В.1-В.53.

Спочатку (рис. 4.1) користувач має обрати приклад: або приклад з неорієнтованим графом, або з орієнтованим. На рис. 4.1-4.28, В.1-В.27 показана робота з першим прикладом (неорієнтованим графом), на інших (рис. В.28-В.53) – з другим прикладом (орієнтованим графом).

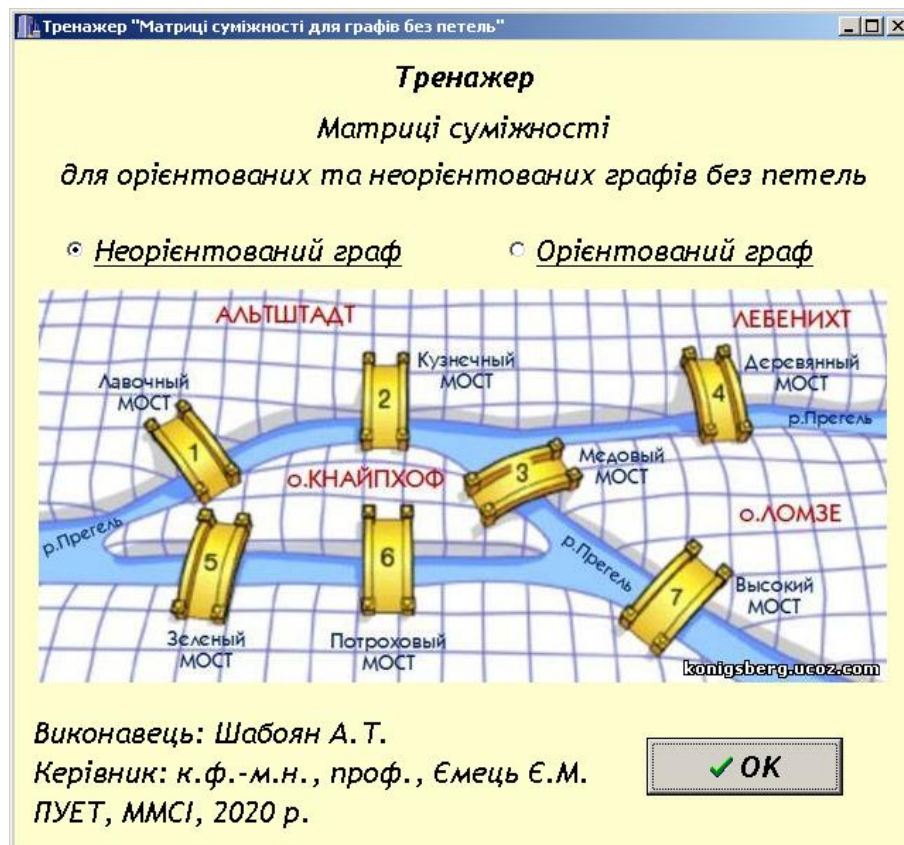


Рисунок 4.1 – Початок

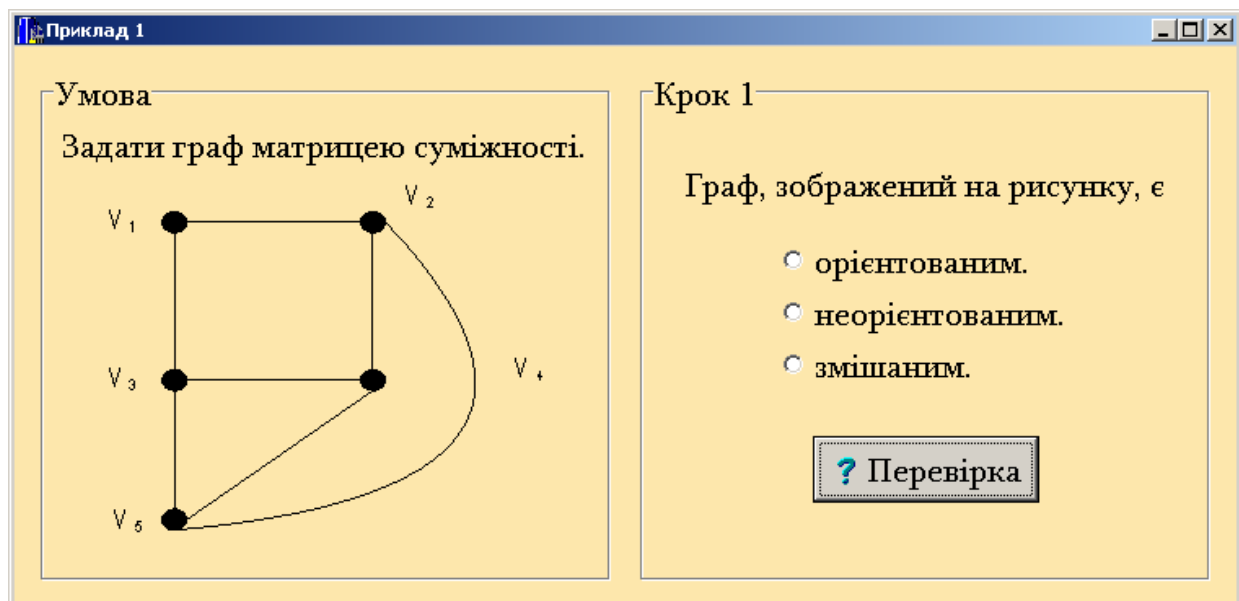


Рисунок 4.2 – Крок 1

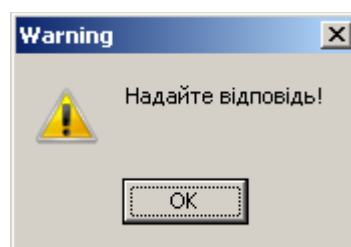


Рисунок 4.3 – Обробка ненаданої відповіді

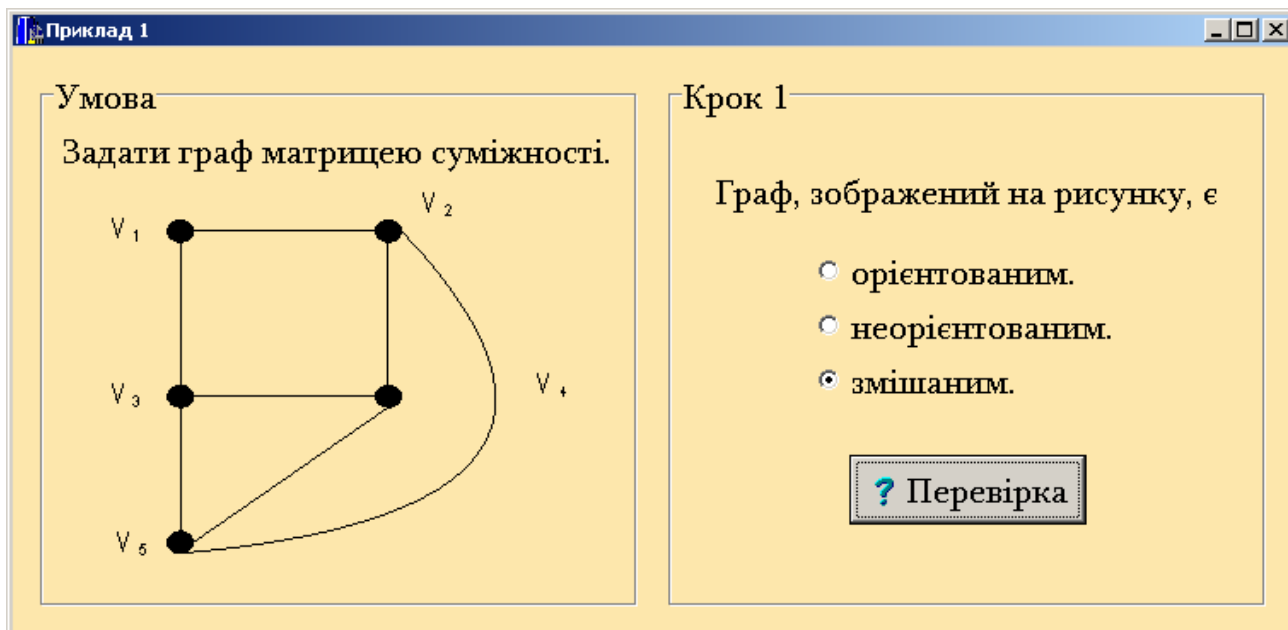


Рисунок 4.4 – Помилка

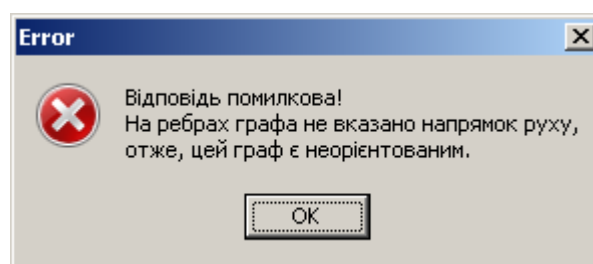


Рисунок 4.5 – Пояснення помилки

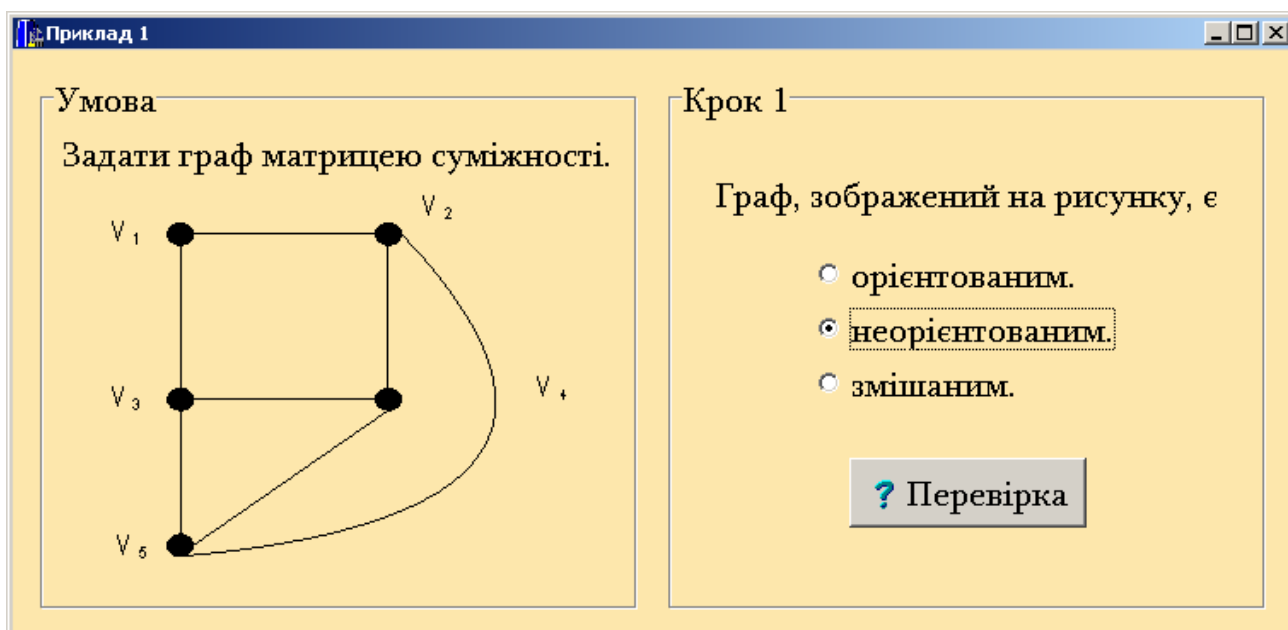


Рисунок 4.6 – Вірна відповідь

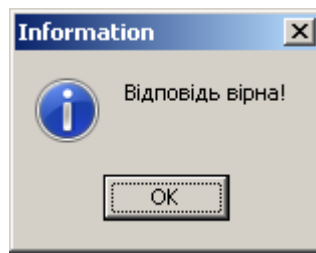


Рисунок 4.7 – Схвалення наданої відповіді

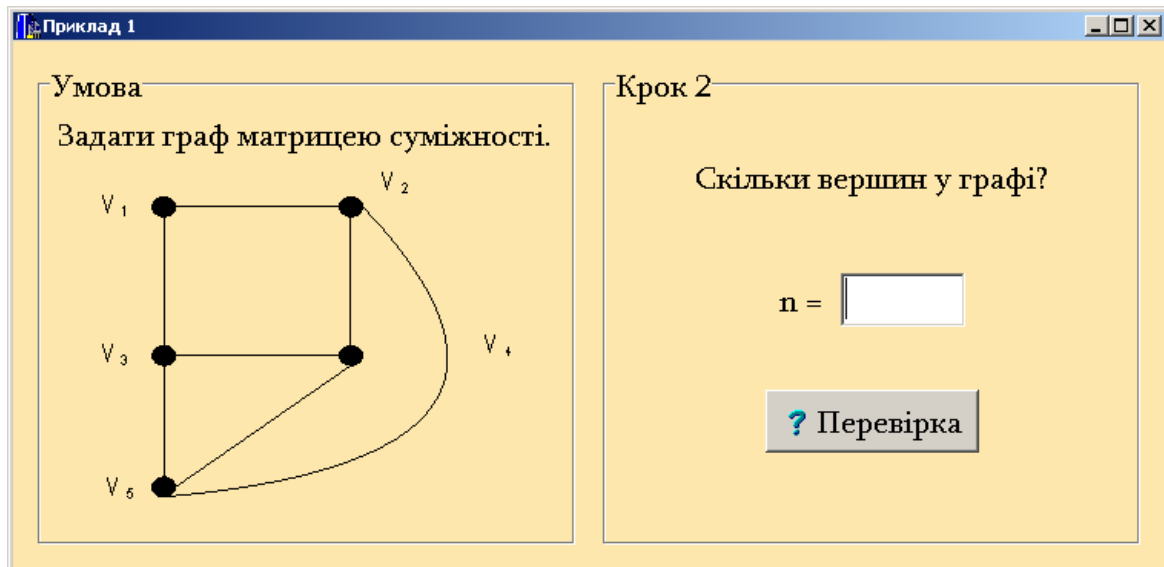


Рисунок 4.8 – Крок 2

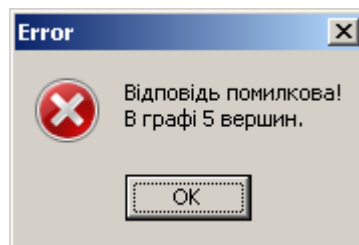


Рисунок 4.9 – Пояснення помилки

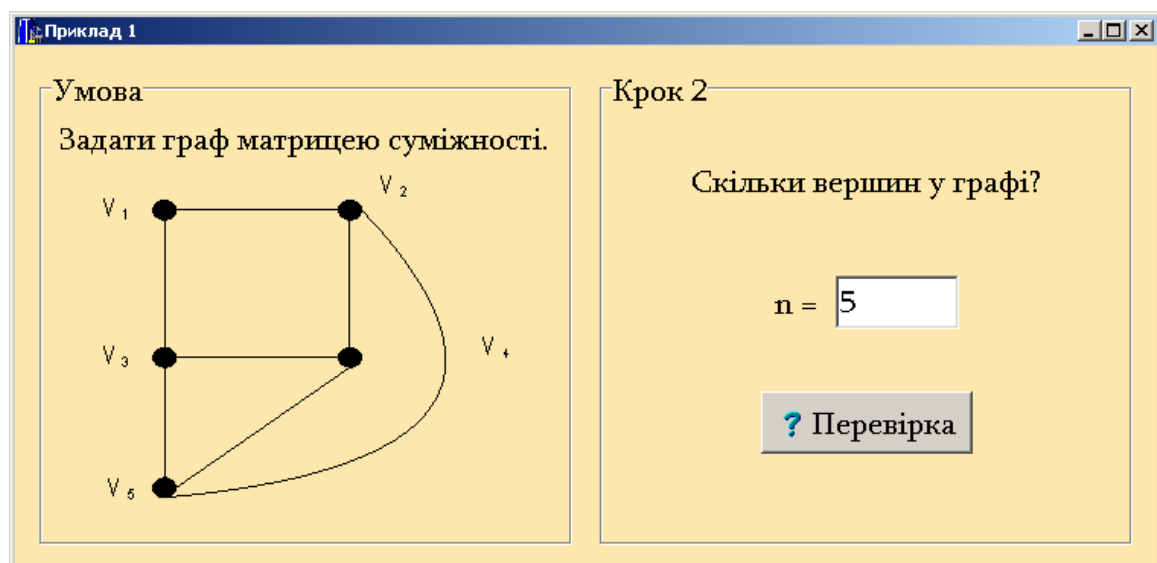


Рисунок 4.10 – Вірна відповідь

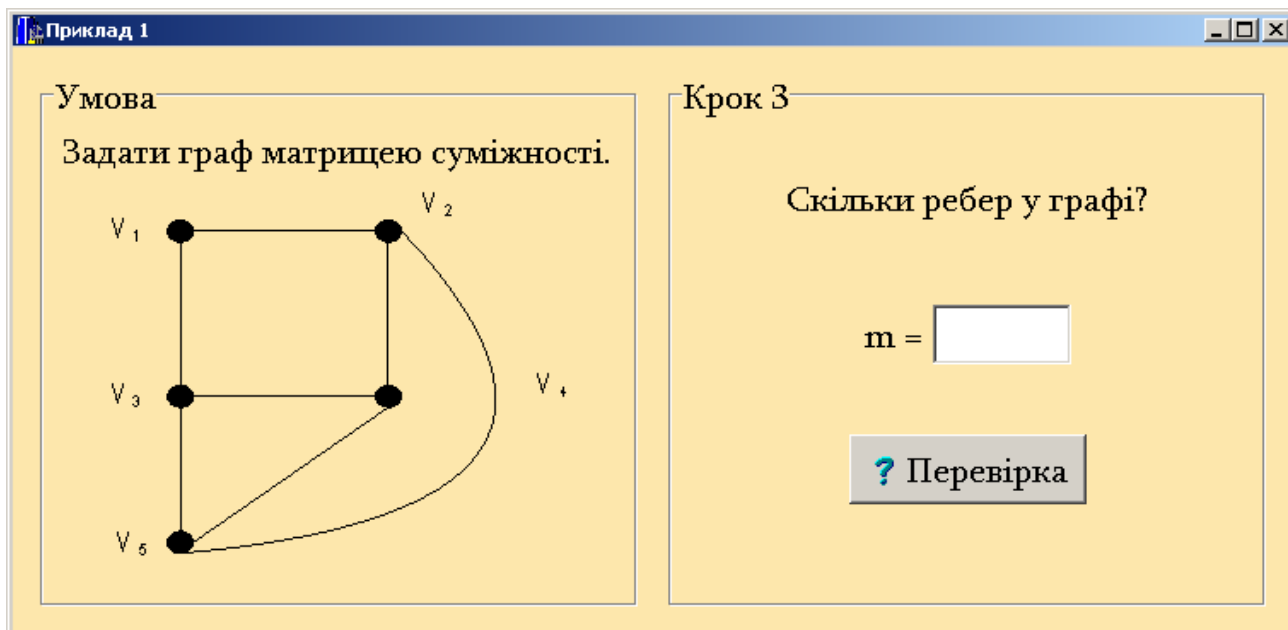


Рисунок 4.11 – Крок 3

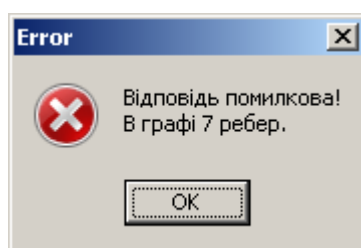


Рисунок 4.12 – Пояснення помилки

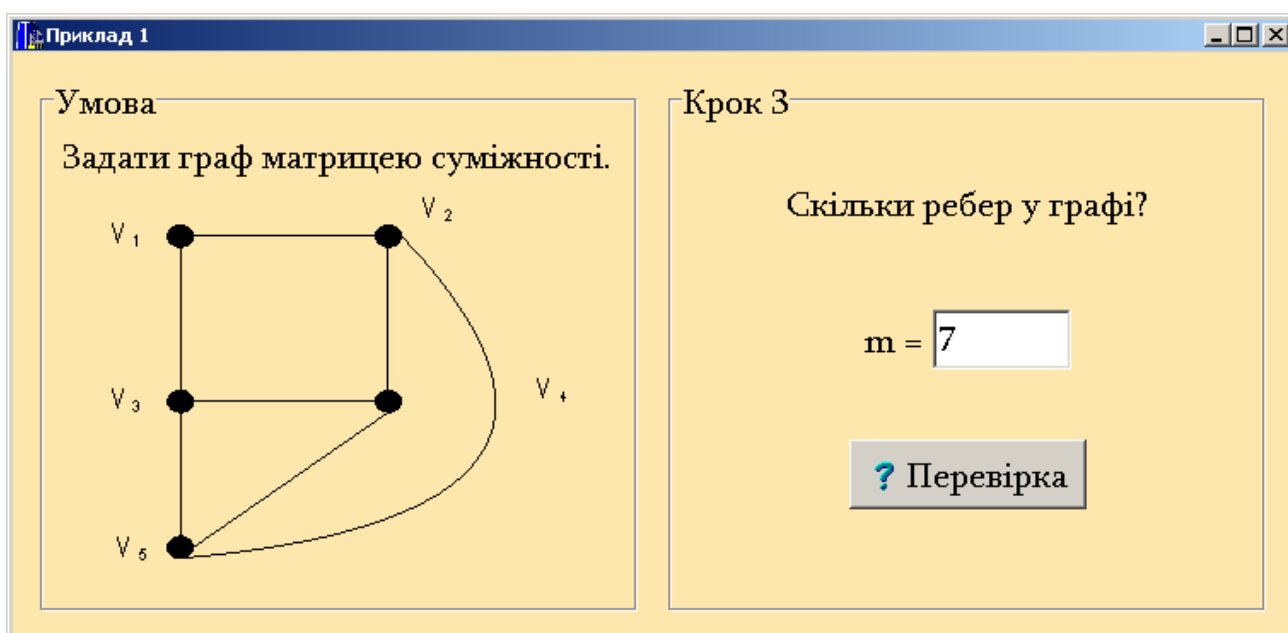


Рисунок 4.13 – Вірна відповідь



Рисунок 4.14 – Крок 4

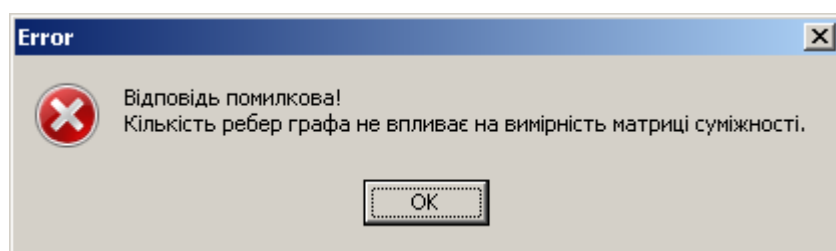


Рисунок 4.15 – Пояснення помилки

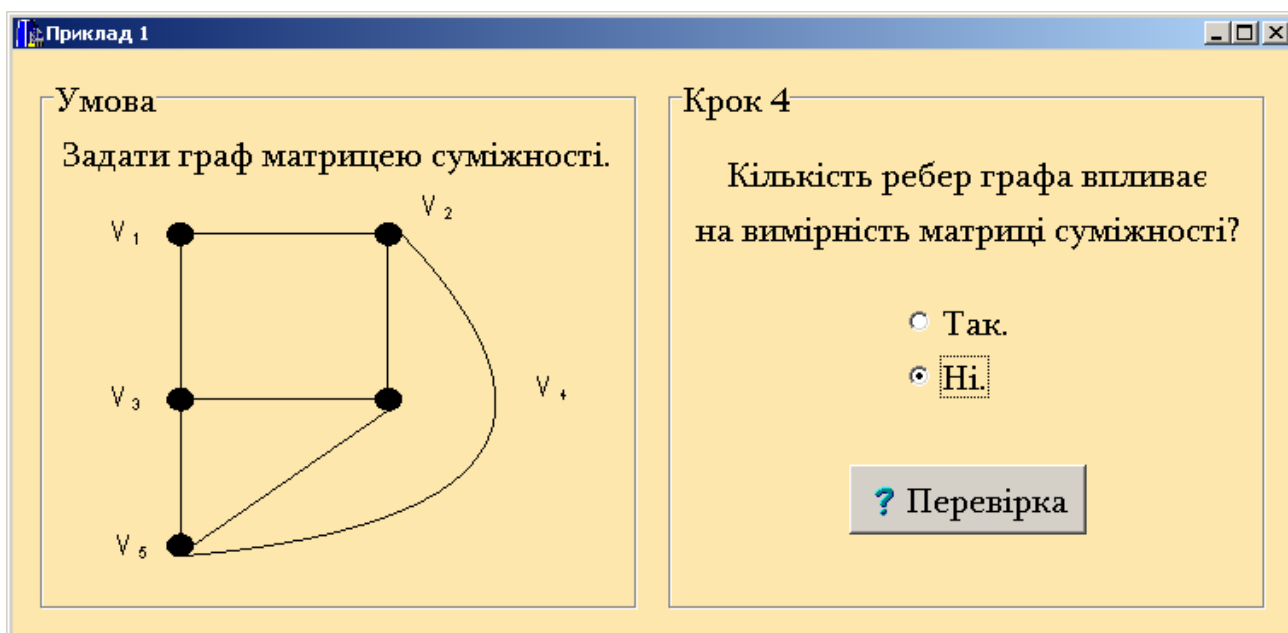


Рисунок 4.16 – Вірна відповідь

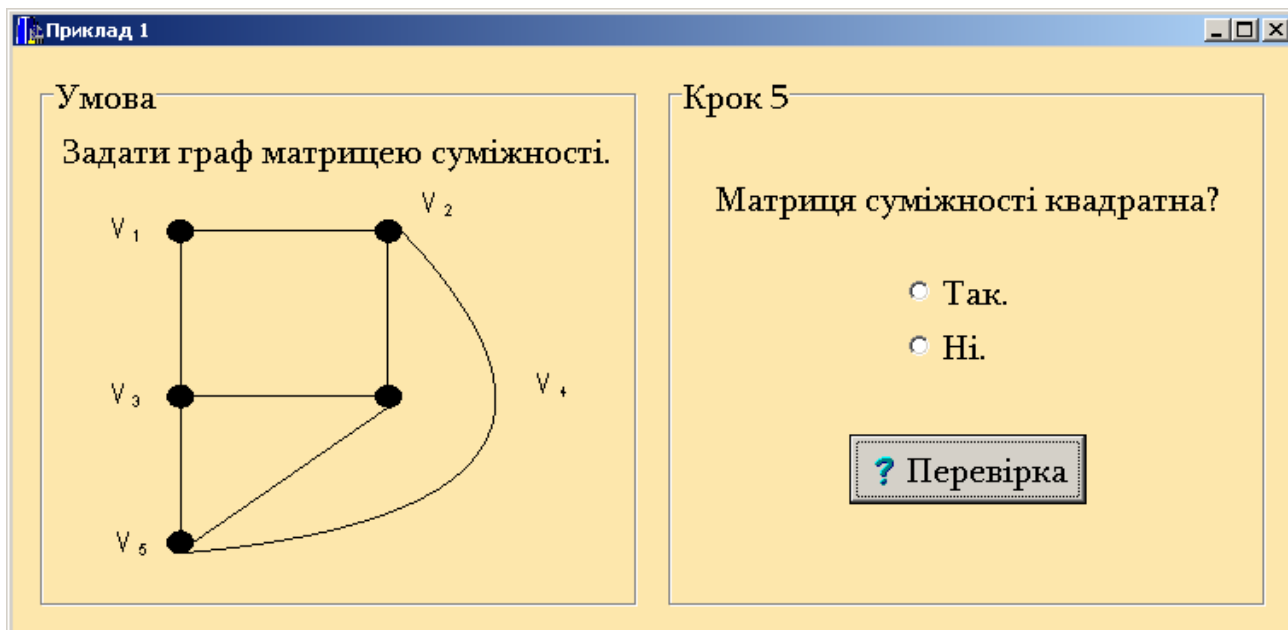


Рисунок 4.17 – Крок 5

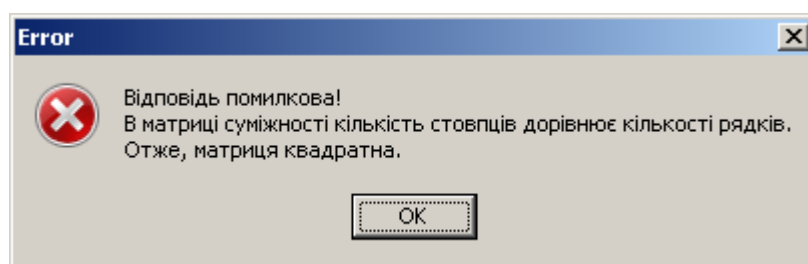


Рисунок 4.18 – Пояснення помилки

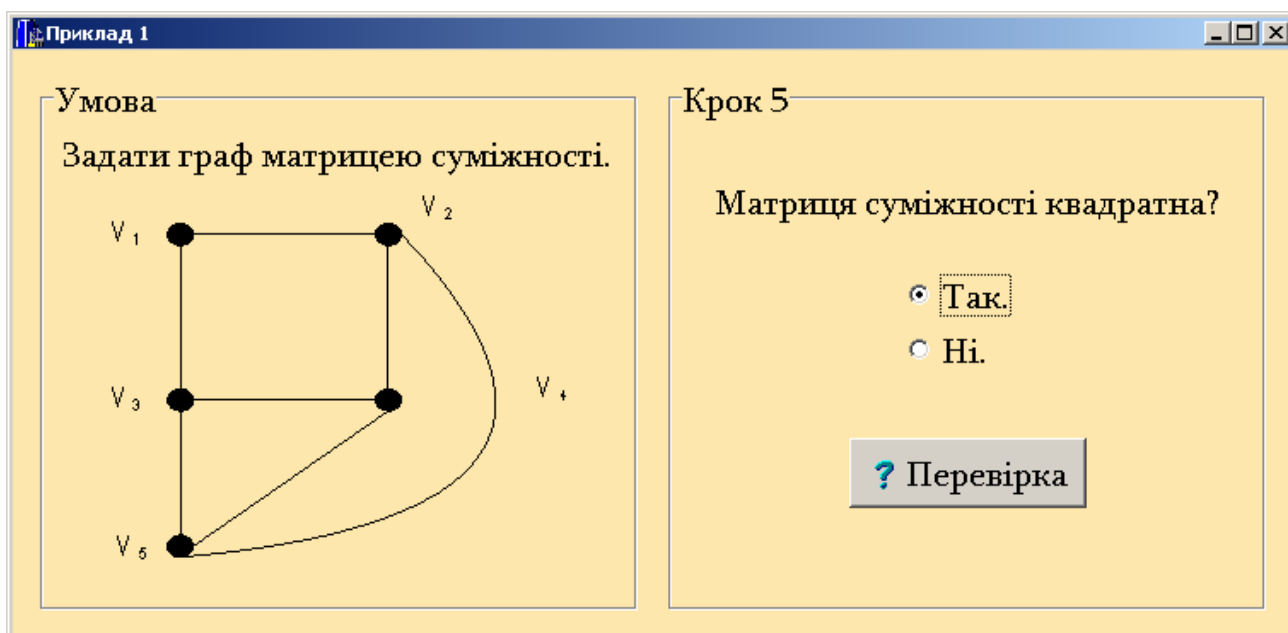


Рисунок 4.19 – Вірна відповідь

Приклад 1

Умова

Задати граф матрицею суміжності.

Крок 6

Яка вимірність матриці суміжності?

x

? Перевірка

Рисунок 4.20 – Крок 6

Error

Відповідь помилкова!
В матриці суміжності кількість рядків і кількість стовпців дорівнюють кількості вершин графа.
Отже, вимірність матриці 5 x 5.

OK

Рисунок 4.21 – Пояснення помилки

Приклад 1

Умова

Задати граф матрицею суміжності.

Крок 6

Яка вимірність матриці суміжності?

x

? Перевірка

Рисунок 4.22 – Вірна відповідь



Рисунок 4.23 – Крок 7

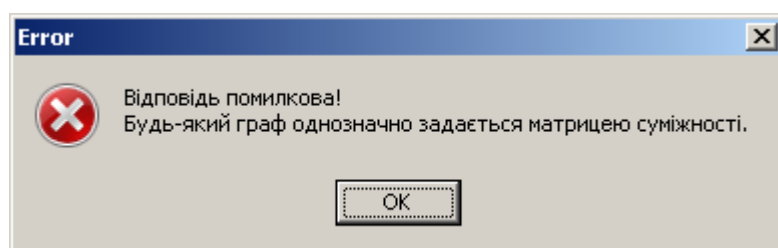


Рисунок 4.24 – Пояснення помилки



Рисунок 4.25 – Вірна відповідь

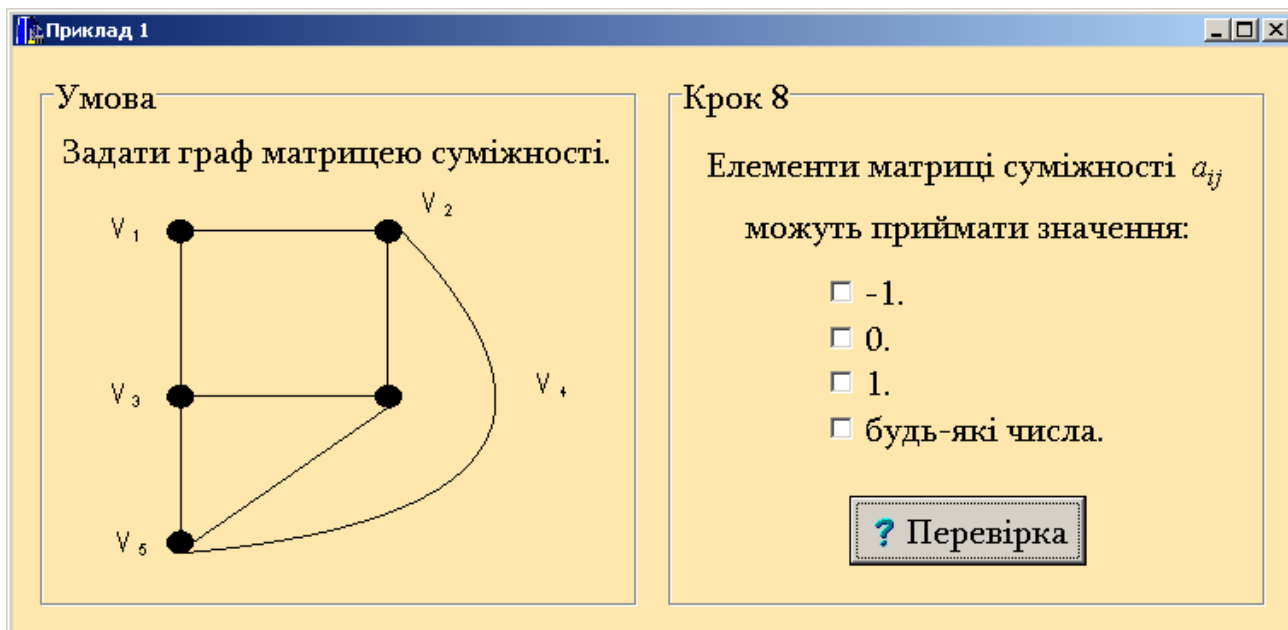


Рисунок 4.26 – Крок 8

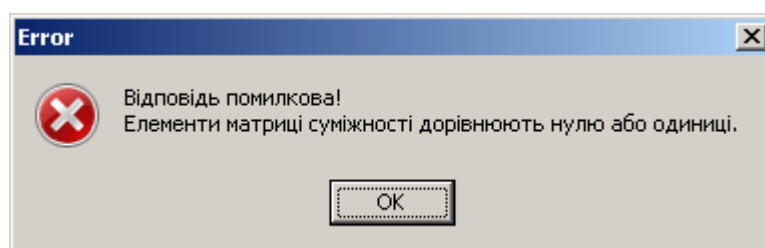


Рисунок 4.27 – Пояснення помилки

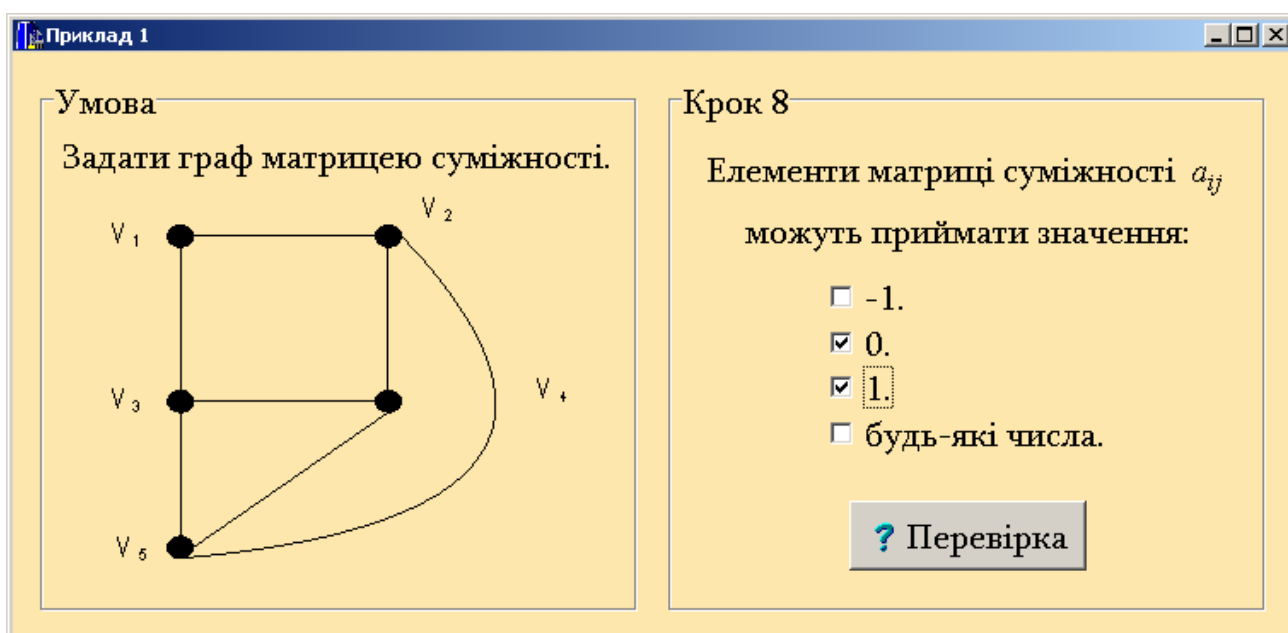


Рисунок 4.28 – Вірна відповідь

ВИСНОВКИ

При виконанні бакалаврської роботи відбулось знайомство з темою «Задання графів матрицями суміжності».

У випусковій роботі розглядалися лише неорієнтовані та орієнтовані графи без петель.

З дистанційного курсу «Алгоритми та структури даних» Полтавського університету економіки і торгівлі було взято два приклади неорієнтованого та орієнтованого графів та побудованих для них матриць суміжності.

Для цих прикладів були створені алгоритми тренажеру. Приклад 1 складається з 15 кроків, а приклад 2 – з 8 кроків.

Було створено блок-схему для першого прикладу для перших дев'яти кроків алгоритму.

Алгоритми було запрограмовано мовою C++ у середовищі програмування Borland Builder 5.

Програма перевірена. Усі помилки усунуті.

Тренажер передано на впровадження у дистанційний курс «Алгоритми та структури даних» Полтавського університету економіки і торгівлі, що засвідчує акт впровадження.

Результати роботи пройшли обговорення на науково-практичних семінарах «Комп'ютерні науки і прикладна математика». Були опубліковані тези [6, 7].

СПИСОК ЛІТЕРАТУРНИХ ДЖЕРЕЛ

1. Ємець О.О. Дистанційний курс ПУЕТ «Алгоритми та структури даних» для студентів спеціальностей «Комп'ютерні науки та інформаційні технології», «Комп'ютерні науки» / О.О. Ємець. – [Електронний ресурс].
2. Соколов О. Ю. Інформатика для інженерів / О. Ю. Соколов, І. Т. Зарецька, Г. М. Жолткевич, О. В. Ярова. – Харків: Факт, 2006. – 424 с.
3. Ємець О. О. Дискретна математика: Навчальний посібник. Вид. 2-ге, допов. / О. О. Ємець, Т. О. Парфьонова. – Полтава: РВВ ПУСКУ, 2009. – 287 с.
4. Івахова Ю.С. Програмне забезпечення для тренажера з теми: «Матриця суміжності та інцидентності» дистанційного навчального курсу «Дискретна математика» / Ю. С. Івахова, Т. О. Парфьонова // Від ефективного управління до ефективної економіки: збірник наук. статей магістрів факультету економіки та менеджменту за результатами наукових досліджень 2012-2013 н.р. – Полтава: ПУЕТ, 2013. – С. 228-232. – Режим доступу: <http://dspace.puet.edu.ua/handle/123456789/1717>
5. Ємець О. О. Методичні рекомендації до виконання бакалаврської роботи для студентів спеціальності 122 «Комп'ютерні науки та інформаційні технології» освітня програма «Комп'ютерні науки» галузь знань – 12 «Інформаційні технології» / О. О. Ємець. – Полтава: ПУЕТ, 2017. – 71 с.
6. Шабоян А. Т. Тренажер «Матриці суміжності для неорієнтованих графів без петель» / А. Т. Шабоян, Є. М. Ємець, О. О. Ємець // Комп'ютерні науки і прикладна математика (КНіПМ-2020): матеріали наук.-практ. семінару. Випуск 5. / За ред. Ємця О. О. – Полтава: Кафедра ММСІ ПУЕТ, 2020. – С. 17-21. – Режим доступу: <http://dspace.puet.edu.ua/handle/123456789/8269>.
7. Шабоян А. Т. Тренажер «Матриці суміжності для орієнтованих графів без петель» / А. Т. Шабоян, Є. М. Ємець, О. О. Ємець // Комп'ютерні науки і прикладна математика (КНіПМ-2020): матеріали наук.-практ. семінару. Випуск 5. / За ред. Ємця О. О. – Полтава: Кафедра ММСІ ПУЕТ, 2020. – С. 52-55. – Режим доступу: <http://dspace.puet.edu.ua/handle/123456789/8905>.